

TARTU ÜLIKOOL

Arvutiteaduse instituut

Informaatika õppekava

Karl-Johannes Kalma

Puude käsitusviisid ja õpetamine

Bakalaureusetöö (9 EAP)

Juhendaja: Ahti Peder, PhD

TARTU 2019

Puude käsitusviisid ja õpetamine

Lühikokkuvõte:

Bakalaureusetöös uuritakse puude õpetamist maailma ülikoolides ning võrreldakse seda Tartu Ülikoolis õpetatavate alateemadega. Töö teoreetilises osas tuuakse välja erinevad õpetamis-metoodikad ning võrreldakse nende erinevusi ja sarnasusi. Viimastes peatükkides antakse ülevaade uurimuse käigus kogutud andmetest, analüüsitakse neid ning sõnastatakse uurimuse tulemused.

Võtmesõnad:

Puu, andmestruktuur, õppematerjalide analüüs, õpetamine

CERCS: P175 Informaatika, süsteemiteooria, S270 Pedagoogika ja didaktika

Teaching and usage of tree data structures

Abstract:

The following thesis focuses on analysing how different universities over the world teach tree data structures and compares the results to exercises and topics used in University of Tartu. Theoretical part of the thesis mainly speaks about different approaches of the topic and compares differences and similarities of those approaches. Final chapters of the thesis analyse the data gathered during the research and gives an overview of the results.

Keywords:

Tree, data structure, analysis of teaching materials, teaching

CERCS: P175 Informatics, systems theory, S270 Pedagogy and didactics

Sisukord

1	Sissejuhatus	4
1.1	Teema tutvustus.....	4
1.2	Töö metoodika	4
1.3	Töö struktuur	5
2	Uurimus.....	6
2.1	Eesti ülikoolide metoodika	6
2.1.1	Eesti ülikoolidele sarnane metoodika	9
2.2	Kordav metoodika	10
2.2.1	Waterloo Ülikool.....	10
2.2.2	Massachusettsi Tehnoloogiainstituut.....	12
2.2.3	Helsingi Ülikool.....	14
2.2.4	Stanfordi Ülikool.....	16
2.3	Kordamiseta kursused.....	17
2.3.1	Edinburghi Ülikool	18
2.3.2	Queenslandi Ülikool.....	19
3	Analüüs.....	21
4	Näidisülesanded.....	25
4.1	Kordavad teemad	25
4.2	Kahendotsimise puud	26
4.3	Tasakaalustatud puud	28
4.4.	Kuhjad	29
4.5	Keerulisemad ülesanded.....	30
5	Kokkuvõte	33
	Viidatud kirjandus.....	34
	Lisad.....	36
	Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks	36

1 Sissejuhatus

1.1 Teema tutvustus

Puu on hierarhiline andmestruktuur, mis koosneb tippudest ja kaartest. Samuti vajab selgitust kahendpuu – tegemist on puuga, mis on kas tühi või milles on üks tipp (juur) eraldatud ja teised tipud on jaotatud maksimaalselt kaheks mittelõikuvaks alamhulgaks, mis on omakorda kahendpuud. [1]

Puid kasutatakse paljudes valdkondades. Näiteks on sugupuud üles ehitatud puustruktuuril, arvutis kasutatakse failide hoiustamiseks kataloogipuud ning erinevatel võistlustel on tihti kasutusel tabel, kus puu ehitatakse üles tavapärasele vastupidiselt, nimelt alustatakse lehtedest, mängu lõppedes liigub edasi vaid üks mängija ja läheb kõrval olevaga kokku ning seda seni, kuni on alles üks liige ehk juurtipp. [2]

Käesolevas töös uuritakse, kuidas õpetatakse puude teemat Tartu Ülikoolis, teistes Eesti ülikoolides ja kursustel üle maailma ning võrreldakse saadud tulemusi. Lõppeesmärgiks on seatud, et valminud tööd põhjal saab lugeja aimu, kuidas õpetatakse puude teemat maailma erinevates ülikoolides ning kuidas on kõige optimaalsem õpetada seda algoritmide ja andmestruktuuride kursuse käigus.

1.2 Töö metoodika

Tartu Ülikoolis õpetavate teemade kohta võetakse materjalid kursuselt „Algoritmid ja andmestruktuurid“ (edaspidi AA). Võrdlusena Eestist kasutatakse nii TalTechi kui ka Tallinna Ülikooli samanimelisi kursuseid. Väljastpoolt Eestit tehakse valik koolide hulgast, mis on pigem maailma reitingutabelis Tartu Ülikoolile lähemal ning mille kursused on tasuta kättesaadavad. Samuti uuritakse läbi paar reitingutabeli tipus olevat ülikooli, saamaks teada, kas nende õppemetoodika erineb millegi poolest, ja kui erineb, siis mille poolest.

Pärast kursuste läbitöötamist analüüsitakse saadud tulemusi ning leitakse vastused küsimustele, mis on kõige levinumad ülesannete tüübid, mis tüübid võiksid anda parima tulemuse õpilastele teema selgestegemisel ning mida peaks Tartu Ülikooli AA kursusel teistmoodi õpetama.

1.3 Töö struktuur

Teoreetiline osa on jaotatud kolme alapeatükki. Esimeses peatükis kirjeldatakse metoodikat, mida kasutatakse Eesti ülikoolides ja tuuakse lisanäidetena välja ülikoolid, kus kasutatakse sarnaseid meetodeid ning võrreldakse neid. Teises peatükis keskendutakse ülikoolidele, kus kasutatakse puude teema õpetamisel teistsuguseid metoodikaid, kuid siiski selgitavad veel oma kursuse käigus kõige algelisemaid teadmisi. Kolmandas peatükis on juttu kursustest, mis eeldavad juba piisavalt eelteadmisi ning keskenduvad peamiselt keerulisemate teemade õpetamisele.

Analüüsi pooles tuuakse täpsemalt välja iga meetodi paremad ja halvemad omadused ning üritatakse välja selgitada kõige optimaalsem õpetamismeetod. Selle analüüsi põhjal pakutakse välja komplekt ülesannetest, mis peaks antud teemat kõige paremini õpetama.

2 Uurimus

Nagu sissejuhatuses mainitud, siis uurimuslik osa on jaotatud kolmeks alapeatükiks, kus selgitatakse lähemalt erinevaid lähenemisviise puude teema õpetamiseks.

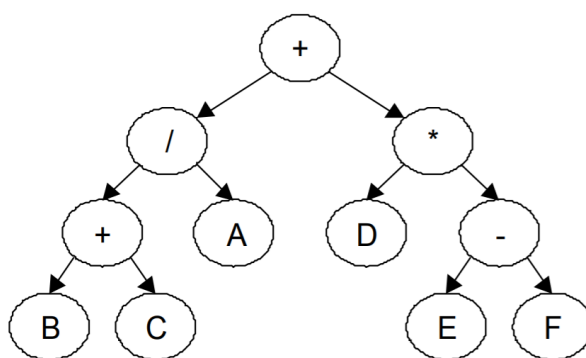
2.1 Eesti ülikoolide metoodika

Uurimuse käigus selgus, et kõik kolm uuritavat Eesti ülikooli (Tartu Ülikool, edetabelikoht 301-350, Taltech, edetabelikoht 601-800 ja Tallinna Ülikool, ei mahtunud edetabelisse) kasutasid väga sarnaseid meetodeid puude teema õpetamisel. Seda võib arvatavasti selgitada sellega, et kõik on oma üheks allikaks lisanud Jüri Kiho poolt kirjutatud raamatu „Algoritmid ja andmestruktuurid“, mille viimane parandatud versioon ilmus 2003. aastal ja sellepärast kattuvad väga mitmed ülesanded ja selgitusviisid kõigis kolmes ülikoolis.

TalTechis ja Tallinna Ülikoolis alustati teemat põhimõistete selgitamisega, kuid Tartu Ülikoolis mainiti kõigepealt ainult mõisteid, ilma milleta ei saaks hakkama ja siis toodi näide puust, mille põhjal selgitati järgmiseid mõisteid. [2-4] Mõlemal metoodikal on omad eelised ja omad miinused. Nimelt, kui õpilane tutvub esimest korda puudega, siis on põhimõistete teadmine väga tähtis, kuid ilma visuaalse selgitusega võib see omandamata jääda, aga samas kui õpilane mõistab teemat, siis näite põhjal selgitamine võtab rohkem aega.

Kõigi ülikoolide materjalides kasutati esimese ülesandena aritmeetilise avaldise puud. Näitena võib tuua TalTechi loengumaterjalist võetud ülesannet, kus tuleb kujutada aritmeetilist avaldist

$$\frac{B+C}{A} + D(E - F) \text{ puuna. [2]}$$

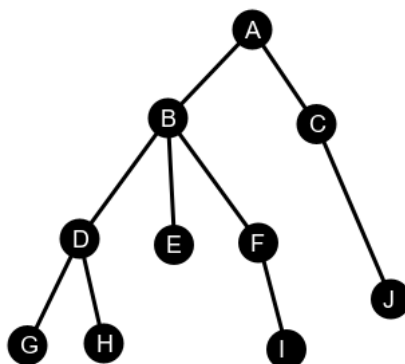


Joonis 1. TalTechi materjalides kasutatav aritmeetilise avaldise puu.

Selle ülesande visuaalne kirjeldus on näidatud joonisel 1, selline puu väljendab antud avaldise tehete prioriteete. Sellise puuga antud avalise väärtuse leidmiseks on vaja osata läbida puud (rekursiivselt) lõppjärjestuses. Seega on selline rekursiivne tehnika ülioluline puude õppimisel.

Materjalis kasutati seda näidet puude läbimise kolme nõ klassikalise järgnevuse (lõppjärjekorra, eesjärjekorra ja keskjärjekorra) õpetamiseks. [2] Selline meetod võib olla sobiv õpilasele, kes on varasemalt puustruktuuriga kokku puutunud, kuid kui õpilane tutvub sellega esimest korda, siis võib teema raskeks jääda. Näiteks on Tartu Ülikooli Arvutiteaduse instituudi poolt välja antud AA ülesannete kogus enne aritmeetilise avaldise puu ülesandeid pühendatud terve peatükk puu läbimisele ehk ülikoolide loengumaterjalides on eeldatud, et õpilane suudab käigult ära õppida mõlemad teemad korraga. [3]

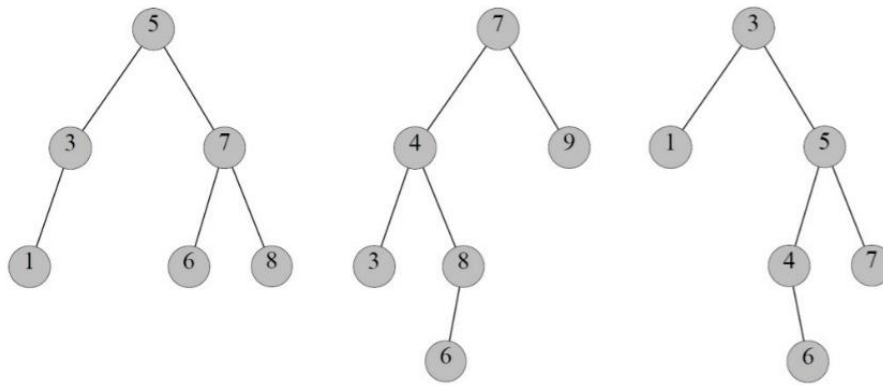
Nii TalTechi kui ka Tartu Ülikooli materjalides oli pärast aritmeetilise avaldise puid lisatud veel selgitavaid näiteid ja ülesandeid järgnevuste kohta. [2-4]



Joonis 2. Tartu Ülikooli näidisülesanne järgnevustest.

Joonisel 2 on näitena toodud üks Tartu Ülikooli näidisülesanne, mille põhjal selgitati järgnevusi (eesjärjestuses on lahendus ABDGHEFICJ, lõppjärjestuses GHDEIFBJCA ja keskjärjestuses, kui puuduks tipp E GDHBFIACJ). Selle selgitamiseks kulutati veel paar lehekülge, mis mahu poolest on u 10% kogu loengust. [3]

Pärast järjestusi tuli teemade järjestuses suurim erinevus. Kui Tartu Ülikool ja Tallinna Ülikool jätkasid puude teemal seletades uusi mõisteid ja teemasid, siis TalTech pühendas järgmised 2 loengut teistele teemadele, kus läbiti graafide teema ja korraldi rekursiooni (mõlemale oli pühendatud üks loeng). Pärast rekursiooni kordamist tulid TalTechi materjalides samad teemad, millega teised ülikoolid oma esimeses loengus jätkasid. Järgmine suurem teema, millel kõik kolm peatusid, oli kahendpuu ja sellega tehtavad operatsioonid (läbimine, lisamine, eemaldamine ning otsimine). [2-4]



Joonis 3. Tartu Ülikooli näidisülesanne kahendotsimise puudest.

Tartu Ülikooli loengumaterjalis oli ülesandena küsitud, milline joonisel olevatest (joonis 3) puudest on kahendotsimise puu omadustega ning vahetult enne oli neid samu omadusi selgitatud. TalTechi ja Tallinna Ülikooli materjalides oli väljatoodud omaduste nimekiri, kuid eraldi näidist juurde toodud ei olnud. Tallinna Ülikool oli eraldi lisanud iseseisva ülesandena morse koodi tõlgendamise, mis oli üles ehitatud kahendpuu põhimõttel ja kasutasid seda täiendava teemaselgitusvahendina. [2-4]

Järgmise teemana võtsid Tallinna ülikoolid kuhja, kuid Tartu Ülikool jätkas kahendotsimise puu erinevate operatsioonidega, mis võimaldas õpilastel kinnistada eelmist teemat ning ehk ka paremini mõista seda. [2-4]

Pärast kahendotsimise puu operatsioonide läbimist võetakse kõigis kolmes ülikoolis käsile puude tasakaalustamine. See teema aitab muuta puudel sooritatavate operatsioonide ajalise keerukuse lineaarselt keerukuselt logaritmilisele, mis suurema andmehulga peal muudab tegevused oluliselt kiiremaks. Tasakaalustamise juures on igas koolis mainitud kõrguse poolest tasakaalustatud puude (edaspidi tasakaalustatud puude) erijuhte ning siis on pikemalt selgitatud erinevaid pöördeid (vasak, parem, paremvasak, vasakparem), mis aitavad hoida puid tasakaalus. Kokku on igas ülikoolis sellele pühendatud umbes üks loeng. Viimase teemana läbitakse Tartu Ülikoolis kuhjad, mida TalTech ja Tallinna Ülikool tegid juba paari teema eest. [2-4]

Kui võrrelda kogu kursuse ülesehitust, siis oli kõigi lähenemisviis teemadele sarnane – läbiti samad teemad, selgitati samu mõisteid ning alustati eeldusega, et inimene ei ole enne puudega kokku puutunud. Kuigi leidis ka erinevusi (näiteks TalTechis oli eraldi rekursiooni loeng puu loengute vahel), siis üldiselt võib öelda, et kõik kolm kooli õpetavad põhimõtteliselt ühtemoodi. Isegi teemadele pühendatud loengute arv oli kõigil kolmel võrdne (3). [2-4]

2.1.1 Eesti ülikoolidele sarnane metoodika

Kui Eesti koolide materjalide sarnasust saab selgitada ühise algallika kasutamisega, siis Aucklandi Ülikooli (edetabelikoht 201-250) õppematerjalid sarnanesid väga Eesti koolide omadele, kuid neil polnud ühtegi ühist allikat. Ainukene sarnasus nendel ülikoolidel esmapilgul oli nende edetabelikoht maailma ülikoolide nimekirjas (mõlemad ülikoolid on esimesest kahe sajast väljas, kuid esimese neljasaja sees). [5]

Kui Eestis läbiti andmestruktuurid ja nende peal rakendatavad algoritmid struktuur haaval, siis selles kursuses läbiti eelnevalt kõik andmestruktuurid eraldi ja siis hiljem nende peal rakendatavad algoritmid, see jagab terve kursuse põhimõtteliselt pooleks ja laseb selgemalt keskenduda ühele teemale, kuid tekib oht, et kui algoritmideni jõutakse, siis vajavad andmestruktuuride omadused kordamist, kuna on juba meelest läinud. [6]

Huvitava kokkusattumisenähtena on selle aine raames kasutatud puude näitena samuti aritmeetilise avaldise puu (siin kujutati avaldist $A((B + C)(D * E) + F)$ puuna), mida kasutasid ka kõik Eesti ülikoolid ja Queenslandi Ülikool. Puude teemas oli sissejuhatus küll lühike, kuid läbiti kõik vajalikud teemad ja mõisted. Järgmistes teemades seletati küllaltki põhjalikult ja samal tasemel kui Eestis. Algoritmide juures läbiti esialgu täpselt samad teemad samas järjekorras kui Tartu Ülikoolis, kuid hiljem kursuse lõpus oli eraldi loeng Huffmani kodeerimise peale, mida näiteks Tartu Ülikoolide materjalis oli mainitud vaid korra. [6]

Kokku oli Aucklandi Ülikoolis pühendatud puude teema õpetamisele ligikaudu 2,5 loengut, mis on võrreldav Eesti ülikoolide loengute arvuga (3) ja läbitud teemad olid väga sarnased ning samas järjekorras. Suurima erinevusena võib välja tuua, et kõigepealt läbiti andmestruktuurid eraldi ja siis hiljem algoritmid. Lisaks eeldati siin, et kursusel õppiv inimene omas juba alguses mingeid elementaarseid teadmisi sellest teemast ja seega päris sissejuhatavaid asju seal eraldi ei selgitatud. [6]

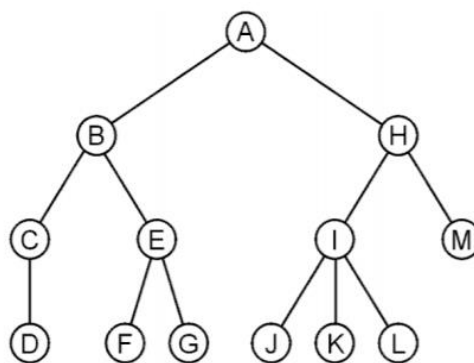
2.2 Kordav metoodika

Selles peatükis on juttu ülikoolide kursustest, kus on käsitletud puude õpetamist mahult umbes sama palju kui Eesti ülikoolides, kuid seda on tehtud minimaalse kordamisega ning pigem on keskendunud uute ja raskemate asjade õpetamisele ja seletamisele.

2.2.1 Waterloo Ülikool

Waterloo Ülikoolis (edetabelikoht 201-250) on kogu kursus jagatud 15 teemaks ja iga teema on omakorda jagatud alateemadeks, millest osad on kohustuslikud ja osad valikulised. Puudele oli kursuses jagatud 3 teemat (puud, tasakaalustatud puud ja otsimispuud), mis on täiesti võrreldav Eesti ülikoolides pakutud mahuga (näiteks Tartu Ülikoolis oli samuti kursuse peale 15 loengut ning puude teemasid oli käsitletud 3 neist). [7]

Esimene teema käsitles puid üldiselt ning oli jagatud 6 alateemaks (sissejuhatus, abstraktsed puud, puu läbimine, vanemale viitamisega puud, metsad ja rekursioon), millest esimesed 3 olid kohustuslikuks märgitud ning viimased 3 olid valikulised. Sissejuhatuses tegeleti põhiliselt erinevate mõistete seletamisega näidete põhjal. Näiteks enamus puu põhimõisteid (nagu leht, juur, tipp jms) seletati näidispuu (joonis 4) põhjal. [7]



Joonis 4. Waterloo ülikooli näidispuu.

Pärast mõistete seletamist toodi elulisi näiteid puudest (nagu HTML dokument). Kõigele sellele oli kokku kulutatud 8 lehekülge, mis on u 40% esimese teema kohustuslikust osast. [7]

Teises alateemas oli esimeses teemas õpetatud teemade praktilisem õpetus. Terve teema oli pühendatud puude erinevate omaduste leidmisele programmeerimiskeeles C++. Näiteks ülesannet, kus seletati puid ja selle hierarhilist korda, selgitati realiseeritud puu abil (joonis 5). [7]

```

#include <algorithm>
#include "Single_list.h"

template <typename Type>
class Simple_tree {
private:
    Type element;
    Simple_tree *parent_node;
    ece250::Single_list<Simple_tree *> children;

public:
    Simple_tree( Type const &, Simple_tree * = 0 );

    Type retrieve() const;
    Simple_tree *parent() const;

    bool is_root() const;
    bool is_leaf() const;
    int degree() const;

    Simple_tree *child( int n ) const;
    int size() const;
    int height() const;

    void attach( Type const & );
    void attach( Simple_tree * );
    void detach();
};

```

Joonis 5. Waterloo Ülikoolis kasutatud koodinäide puust.

Selline lähenemisviis oli ainulaadne võrreldes selles lõputöös uuritud teiste kursustega. Tegemist on ainukese kursusega, kus oli teemat loengus seletatud koodinäidetega (TalTechi materjalides oli korra kasutatud koodilõiku Java keeles, kuid see ei olnud korduv nähtus ja oli pigem kasutatud näitena, mitte teema selgitamiseks kasutatava abivahendina). See alapeatükk oli 6 lehekülge pikk ehk 30% kogu esimesest kohustuslikust teemast. Keskenduti eelkõige kõige algelisematele puude funktsioonidele (mingi kindla tipu või puu kõrguse leidmine jms). Viimase kohustusliku teemana läbiti selle peatükki all puude läbimise teema. Sarnaselt esimese teemaga seletati alguses teooriat, toodi näiteid ja kirjeldati lahenduskäike, kuid pärast teema õpetamist oli teemas kasutatud kordavate näidetena taas C++-keeles koodinäiteid ülesannete lahendustest, mis peaks õpilastele andma parema praktilise arusaama teemast. [7]

Erinevalt teistest kursustest oli selles valikulised teemad, mida õpilane võis läbida. Suurem osa nendest teemadest oli näiteks Tartu Ülikoolis või Tallinna Ülikoolis kohustuslike materjalide all ning eriti imekspandav on fakt, et ühe lisateemana on selles kursuses rekursiooni õpetamine, mis peaks olema puude läbimise juures väga oluline oskus, kuid selles kursuses oli peetud seda kõrvaliseks teemaks. Arvatavasti oli rekursiooni käsitletud juba eelmise aine käigus piisavalt põhjalikult, et eeldada õpilastelt piisavalt head teema tundmist. [7]

Teise teemana oli Waterloo Ülikoolis käsitletud tasakaalustatud puid, mis Tartu Ülikoolis ja TalTechis näiteks oli jäetud viimaseks teemaks. Anti teoreetiline ülevaade teemast ning nagu sellele kursusele kohane, siis oli näiteid selgitatud ka C++-keeles koodilõikudega. Kui näiteks kolmes uuritud Eesti ülikoolis oli tasakaalustatud puude teemaks läbitud erinevad tasakaalustamise operatsioonid nagu lisamine, eemaldamine ja pöörded, siis selle kursuse käigus ei olnud neid veel õpetatud ning kogu see teema jäi pigem teooria juurde ja lihtsalt selgitas tasakaalustatud puude omadusi. Alles viimase teema juures (milleks oli otsimispuud) seletati nende peal sooritatavaid operatsioone. Lisaks erinevatele otsimispuudele korraldati kohustusliku materjali hulgas ka puude läbimist eesjärjestuses. Valikuline materjal koosnes erinevatest puude liikidest, mida kajastasid teised kursused oma materjalides vähemal või rohkemal määral. [7]

Selle kursuse suurimaks omapäraks võib pidada teooria selgitamist läbi koodilõikude ja praktiliste näidete. Pärast esimese teema esimest alateemat oli peaaegu igas esitluses vähemalt üks koodinäide. Lisaks sellele kursusele kasutati koodilõike ka Helsingi Ülikoolis ja Stanfordi Ülikoolis (TalTechis oli üks Java keeles olnud näide, kuid selle põhjal ei selgitatud, see oli lisaks juures), kuid sellel kursusel kasutati seda kõige rohkem, väga suur osa õpetamisest oli koodipõhine. Teise erinevusena võib välja tuua ka teemade teistsuguse järjekorra. Viimasest kolmest teemast kaks viimast olid näiteks Eesti ülikoolidega võrreldes vahetatud järjekorras. Siiski ei saa öelda, et kursused oleksid kategooriliselt erinevad. Siin oli samuti seletatud mõisteid eraldi lahti ja oli räägitud ka triviaalsematest teemadest. Kuigi tutvustavate teemade materjalid olid vastavalt 1,5 ja 2,5 korda lühemad (puude teema oli 20lk, tasakaalustatud puude oma 30lk ja otsimispuude teema u 50 lk), siis see oli piisav, et ka eelnevate teadmisteta õpilane suudaks teema omandada selle kursuse käigus ilma, et maha jääks. [7]

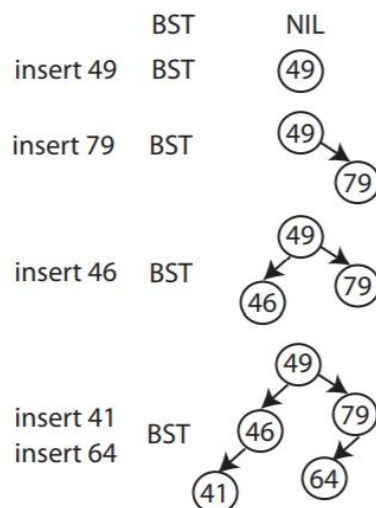
2.2.2 Massachusettsi Tehnoloogiainstituut

Massachusettsi Tehnoloogiainstituudis (edaspidi MIT, edetabelikoht 4) oli algoritmide kursus jagatud 8 teemaks, mis kokku koosnesid 24 loengust. Puude teemat käsitleti 2. teemas ning sellele oli pühendatud 4 loengut. Kuigi kursust nimetatakse sissejuhatuseks algoritmidesse, siis kirjeldusest leiab, et lisaks algoritmidele kaetakse selle käigus ka andmestruktuurid, mida kasutatakse nende algoritmide lahendamiseks. [8]

Teise teema pealkirjaks on sorteerimine ja puud ning see koosneb 5 loengust, millest 4 on puudele pühendatud. Esimene loeng selles teemas oli ülevaade paarist sorteerimismeetodist. Kui võrrelda seni kajastatud teiste kursustega, siis alustab see kursus puude seletamist

kuhjades, ei tehta eelnevat seletustööd puude omadustest ega ei korrata mõisteid. Kuhjade teemat õpetades eeldati ka eelnevaid teadmisi teemadest nagu kahendpuu (kuhja omaduste seletamisel selgitati, et kuhi on massiiv, mis on visualiseeritud peaaegu täieliku kahendpuuna), mis tähendab, et enne seda kursust on antud teemasid kuskil juba käsitletud ja kergemad teemad on eelnevalt selgeks tehtud, siin alustati kohe keerulisemate teemadega. Kuhjade selgitamisel selgitati ka kõige põhilisemad mõisted üle ja läbiti kogu see teema samas mahus nagu näiteks Tartu Ülikoolis seda tehti. [8]

Edasi võeti järgmise teemana kahendotsimise puud. Siin oli samuti eeldatud, et lugeja oli teadlik kahendpuudest, kuid kahendotsimise puu omaduses olid kõik eraldi välja toodud ja teemat selgitati algusest peale väga põhjalikult. Lisaks omadustele olid selgitatud ka nende puude peal tehtavad operatsioonid (nagu lisamine ja otsimine) ning oli selgitatud nende ajaliskerukust. Seda teemat õpetati siin nagu võetakse antud teemat esimest korda ja tehti seda sarnaselt Eesti ülikoolidele. [8]



Joonis 6. MIT materjalides kasutatud näidisülesanne kahendotsimise puust.

Joonisel 6 võib näha selle teema esimest näidisülesannet, kus tühja kahendotsimise puusse hakati ükshaaval lisama kirjeid. Pärast iga kirje lisamist on näidatud puu hetkeseis. Seda näidet kasutati puude omaduste seletamisel ja näide oli materjalis toodud kohe teema alguses, enne kui täpsemalt teemast räägiti. [8]

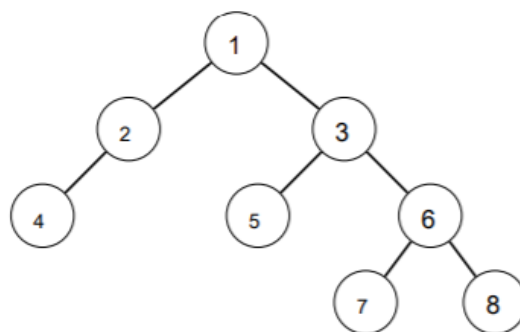
Kolmandaks puude teemaks oli valitud tasakaalustatud puud. Kuna eelmise teema lõpus oli räägitud probleemist, mis võib tekkida kirjete lisamisel suvaliselt (puu sügavus läheb liiga suureks), siis selle teema alguses seletati täpsemalt, miks on puude tasakaalustamine oluline.

Esimese tasakaalustatud puuliigina käsitleti AVL-puid, selgitati kuidas sinna lisada ja eemaldada kirjeid ning kuidas pärast neid toiminguid puu uuesti tasakaalustada. Räägitud oli kõigist pöörete variantidest ja iga kohta oli olemas ka näide, kuidas need toimivad. Kui AVL-puud olid läbitud, siis oli mainitud lisaks muid tasakaalustatud puude liike, kuid lähemalt ühtegi teist enam ei käsitletud ja sellega lõpetati teema. Viimase teemana võeti läbi erinevaid raskemaid sorteerimisalgoritme, mida puudel rakendati (esimeses teemas võetud sorteerimisalgoritmid ei vajanud puude kasutust). [8]

Selle kursuse omapära võrreldes teiste kursustega seisnes kordamise minimaalses olemasolus. Kuigi teemad, mida siin käsitleti, seletati põhjalikult ära ning toodi iga teema kohta ka näiteid, siis sellel kursusel eeldati eelnevaid baasteadmisi puudest ja seda sellises hulgas, et teemat täiesti mitte tundev lugeja ei suudaks arvatavasti materjali ilma lisateadmisi otsimata läbida. Tegemist oli MIT esimese algoritmide kursusega seega sellele eelnev eeltöö oli tehtud mingis eelnevas aines. Kokku pühendati selles aines puude õpetamisele 4 loengut. [8]

2.2.3 Helsingi Ülikool

Helsingi Ülikooli (edetabelikoht 99) kursuse nimetus on Andmestruktuurid ja algoritmid ning selle aine jaoks on valminud samanimeline raamat. Viimane versioon on avaldatud 2019. aasta jaanuaris seega aine materjal on teiste uuritavate kursustega kõige uuem. Selle kursuse puude teema algab küll kahendotsimise puudega, kuid selle teema alguses siiski korratakse vähesel määral üle, mis on tavalised kahendpuud. Korrates kahendpuid, selgitatakse ka puude olemusi, kuid mainitakse, et igal tipul võib olla maksimaalselt kaks alamat. Selgitamata jäävad puud, millel on rohkem kui kaks alamat ja metsad. [9-10]

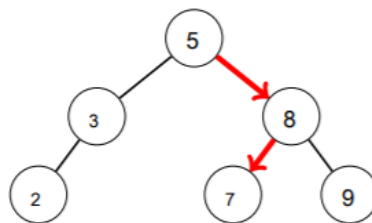


Joonis 7. Helsingi Ülikoolis kasutatud näide kahendpuust.

Joonisel 7 on toodud Helsingi Ülikooli materjalis kasutatud näidis kahendpuust, mille põhjal selgitatakse kahendpuu omadusi. Väga mitmes teises uuritud kursuses oli selleks otstarbeks valitud mingi raskem ülesanne (näiteks Eesti ülikoolides kasutati aritmeetilise avaldise puud),

kuid siin on selleks kõige lihtsamat liiki kahendpuu. Arvatavasti on see sellepärast, kuna puude kordamise teema oli minimaalne ja siis valiti selle võrra kergem näide. Pärast seda oli veel juttu kahendpuu kõrgusest ja tippude arvust ning siis järgmise alateemana võeti läbi, kuidas kahendpuud läbida (kõik kolm järjestust) ja selle omadusi (kõrgust ja tippude arvu) leida, nende kõrvale oli toodud paar näidet kasutades pseudokoodi ning sellega kordav teema lõppes. [9-10]

Edasi võeti kahendotsimise puude teema, alustati puude üldise selgitusega, omaduste kirjeldusega ja räägiti nende vajalikkusest. Pärast lühikest selgitust mindi kohe edasi kahendotsimise puul rakendatavate operatsioonide (nagu otsimine, lisamine ja eemaldamine) juurde. [9-10]



Joonis 8. Helsingi Ülikooli näide kahendotsimise puust otsimisest.

Iga operatsiooni puhul oli alguses teoreetiline selgitus, kuidas asi töötab ning hiljem oli pildiga visuaalne selgitus. Joonisel 8 on näha, kuidas kahendotsimise puust, mille elementide kirjed on numbrid 2, 3, 5, 7, 8 ja 9, otsitakse elementi, mille kirjeks on number 7. Läbiti põhjalikult kõik operatsioonid ning nende erandjuhud (näiteks kui eemaldamisel peab tegema asenduse). [9-10]

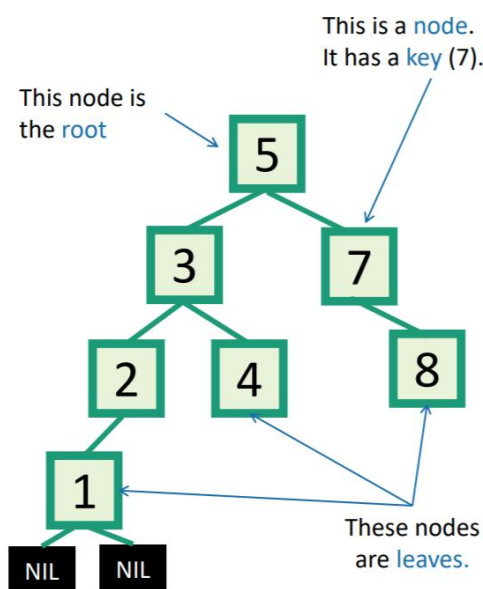
Pärast kahendotsimise puude teemat hakati õpetama tasakaalustatud puid, näiteks Tartu Ülikoolis õpetati teemasid samas järjekorras. Selgitati alguses, miks on vaja tasakaalustada ning seejärel võeti läbi puude peal sooritatavad pöörded ja kuidas neid rakendada. Selleks oli kasutatud ka ühte näidisjoonist, kus visuaalselt näidati ette, kuidas sooritatakse paremvasakpööre ning seejärel mindi teemaga edasi. Kuna kursus keskendub algoritmide ja andmestruktuuride õpetamisele eelkõige Java keeles, siis on eraldi alapeatükk Javas olevatele struktuuridele, mis kasutavad puid oma realisatsioonis (siin võeti läbi Java klassid *TreeSet* ja *TreeMap*). Lõpuks võetakse peatükk kokku võrreldes erinevaid andmestruktuure puudega ja uurides nende ajalist keerukust erinevatel operatsioonidel. [9-10]

Üldiselt oli Helsingi Ülikoolis eeldatud, et seda kursust läbiv inimene on eelnevalt juba puudega mingil määral kursis (kuna kordamisele oli pühendatud pigem väiksem osa ja seal ei olnud läbitud kõiki teemasid), kuid enne uute teemade võtmist oli siiski päris põhjalikult seletatud lahti, mida peaks eelnevalt juba teadma. [9-10]

2.2.4 Stanfordini Ülikool

Stanfordini Ülikooli (edetabelikoht 3) uuritud kursuse nimi on algoritmide disain ja analüüs. Selle kursuse nimi on küll erinev võrreldes teiste uuritud kursustega, kuid sisu on põhimõtteliselt sama, mis üldiselt algoritmide ja andmestruktuuride kursuse käigus käsitletakse. Aine koosneb kokku 17 loengust, millest üks on pühendatud puude teemale. [11]

Sarnaselt Helsingi Ülikoolile alustab Stanfordini Ülikool puude seletamist kahendotsimise puude teemast ning täpselt samamoodi seletab enne otsimispuude teema juurde minemist ära põhimõisted, mida õpilane peaks enne juba teadma. Joonisel 9 on näide selgitavast joonisest, mida kasutatakse selle kursuse käigus sissejuhatavate mõistete kordamiseks. [11]

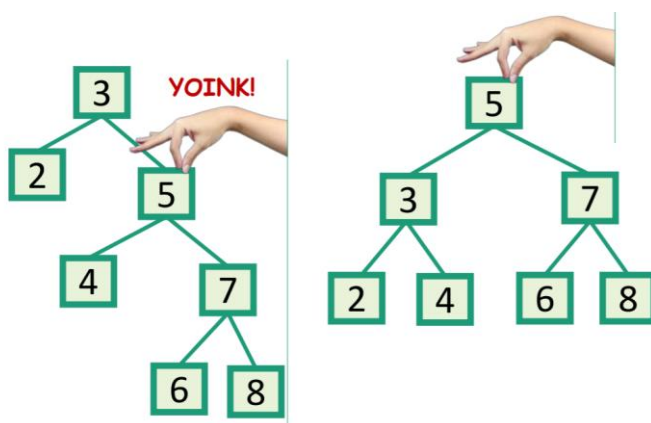


Joonis 9. Stanfordini Ülikooli näide kahendotsimise puust.

Kordamise osa on lühike, kui võrrelda seda näiteks Helsingi Ülikooli kursusega. Korra käiakse mõisted üle koos kõrval oleva näitega ning siis minnakse edasi kahendotsimise puu juurde. Selle teema juures peatutakse pikemalt, alguses seletatakse selle andmestruktuuri omadusi ning kuidas ehitada sellist puud (väiksemad liikmed vasakuks alluvaks ja suuremad paremaks). Pärast seda õpetatakse, kuidas läbida kahendotsimise puud eesjärjestuses ning räägitakse selle peal sooritatavatest operatsioonidest (teisi järjestusi ei mainita). Otsimise, sisestamise ja kustutamise selgitamiseks kasutatakse pseudokoodi näiteid, mis selgitavad, kuidas neid operatsioone teha samm sammu haaval. Kustutamisel peatutakse veidi pikemalt, kuna selgitatakse ära selle kõik võimalikud erijuhtumid. Kahendotsimise puude lõpuks räägitakse

nende operatsioonide ajalisest keerukusest ning selgitatakse, miks on vaja hoida puude kõrgus võimalikult väike. [11]

Kohe pärast näidet räägitakse tasakaalustatud puudest ning kuidas neid tasakaalus hoida. Pöörde õpetamiseks oli vaid kaks pilti, kus oli puudel sooritatud pöörded. Joonisel 10 on kujutatud teine näide sellest, kuidas seletati pöörded kahendpuul. Tasakaalustatud kahendotsimise puu näiteks kasutati selles kursuses puna-must puud. Puna-must puu peal sooritatavaid operatsioone (lisamine, eemaldamine, tasakaalustamine) selgitati kuni loengu lõpuni. [11]



Joonis 10. Stanfordi Ülikooli näide tasakaalustamisel tehtavast pöördest.

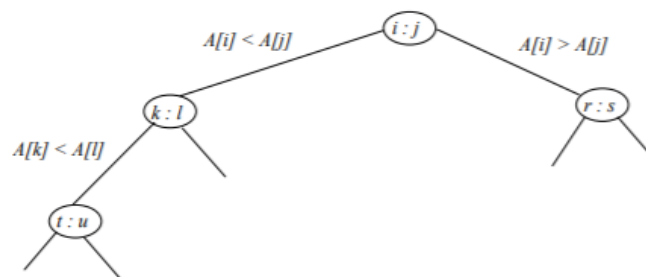
Üldiselt oli selles kursuses puudest vähe juttu. Alustati, sarnaselt mitmele eelnevalt uuritud kursusele, kahendotsimise puudest. Kuigi siin oli väike osa ka kordamiseks, siis sellest pigem mindi ruttu üle ning põhjalikum selgitamine algas hilisemate temade juures. Aine käigus selgitati puudest vaid kahendotsimise puud ja nende tasakaalustatud variante. Arvatavasti võib vähest selgitamist seletada sellega, et see aine on eeldusaineks ühele ainele, mis keskendub vaid andmestruktuuridele. Huvitaval kombel ei olnud selle aine eeldusaineks ühtegi ainet, mis oleks andmestruktuure juba käsitlenud seega võib eeldada, et õpilastelt oodatakse juba eelteadmisi puudest või kiiret arusaamist nendest. [11]

2.3 Kordamiseta kursused

Selles peatükis tuleb juttu ülikoolide kursustest, kus puude kajastus materjalides on minimaalne ja kordamist ei tehta nende raames kas üldse mitte või väga minimaalselt. Põhjuseid nende mittekordamiseks on erinevaid, näiteks keskendutakse ühes kursuses põhiliselt algoritmidele ja sellepärast jäetakse andmestruktuuride osa tahaplaanile.

2.3.1 Edinburghi Ülikool

Edinburghi Ülikooli (edetabelikoht 29) kursuse nimetus on algoritmid ja andmestruktuurid, kuid esimese loengu esimese asjana on mainitud, et antud kursusel keskendutakse pigem algoritmide osale kui andmestruktuuride omale. Seda on kohe ka aru saada, kuna puude osa selles kursuses on minimaalne. Esimest korda mainiti puid sorteerimise osas, mida tehakse väga mitmes kursuses varemgi, kuid selle kursuse teeb eriliseks see, et siin rakendatakse sorteerimisalgoritmi puul enne kordagi selgitama, mis on puu ega öelda, kuidas seda tegema peaks. On eeldatud, et lugeja mõistab seda iseenesest ja on jätkatud lihtsalt algoritmi selgitamisega. [12]



Joonis 11. Edinburghi Ülikooli näide vahelepanemisega sorteerimisest.

Näiteks on joonisel 11 näidatud esimene näide, kuidas seletati vahelepanemisega sorteerimist. Ette oli öeldud, et kahte samasugust kirjet ei esine ehk algoritmis kasutatakse kahendpuu andmestruktuuri, seega on eeldatud, et lugeja mõistab, mis on puu, mis on kahendpuu ja kuidas need töötavad. Tööpõhimõtteid on siin eraldi ka seletatud, kuna algoritmi seletamiseks on uuesti mainitud, mis kirjed kuhu minema peaksid. Pärast vahelepanemisega sorteerimise õpetamist on juttu kiirsorteerimisest ning sellega see peatükk lõppeb. Tegemist oli 7. loenguga 18 loengust ning järgmises loengus seletati kiirsorteerimist lähemalt, kuid puid polnud enam eraldi mainitud. Järgmine kord, kui materjalides mainiti puid, oli 14. loengus, mis on juba graafide osa ja juttu tuli graafi toespudest, mis igas eelnevalt uuritud kursuses oli ka jäetud graafide teemasse. [12]

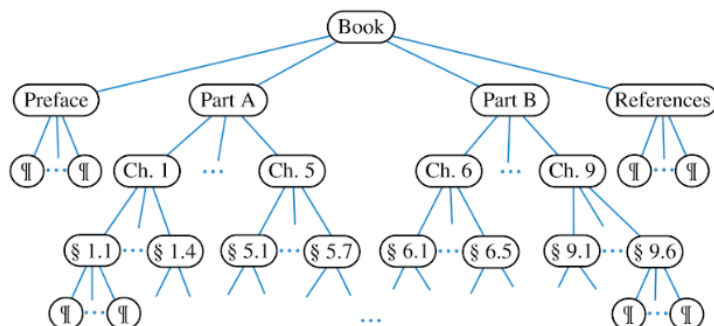
Selles kursuses oli väga selgelt aru saada, et puude andmestruktuuridele ei keskendutud üldse ja eeldati lugejalt piisavaid eelteadmisi. Kuigi algusest peale oli mainitud, et rõhk on pigem algoritmidel kui andmestruktuuridel, siis selle kursuse käigus ei käsitletud ühtegi puudel rakendatavat algoritmi peale sorteerimiste ja toespude konstrueerimise (viimane neist on juba graafide teema). Seda saab seletada sellega, et soovitatud eeldusaineks on sellele ainele pandud „Informatics 2B“, mis annab eelteadmised mõnest algoritmist ja andmestruktuurist, kuid samuti keskendub ka andmetöötlusele ja tehisintelligentsile ehk ei ole otseselt vaid algoritmide ja

andmestruktuuride ettevalmistuseks. Selles aines puude teemat käsitleti läbi AVL-puude, puude põhimõisteid ja omadusi ei korratud, alustati kohe otsimispuudest ja nende tasakaalustamisest sarnaselt MIT kursuse 3. ja 4. loengule, kuid siin oli see ühte loengusse kokku pandud. 3 loengut hiljem oli pool loengust räägitud ka kuhjadest, kuid sellega ka piirdus selle aine käsitus puudest. Aines „*Informatics 2A*“ oli põhiliseks teemaks keeletehnoloogia, seega enam eelnevaid algoritme sisaldavaid aineid polnud. [12-13]

2.3.2 Queenslandi Ülikool

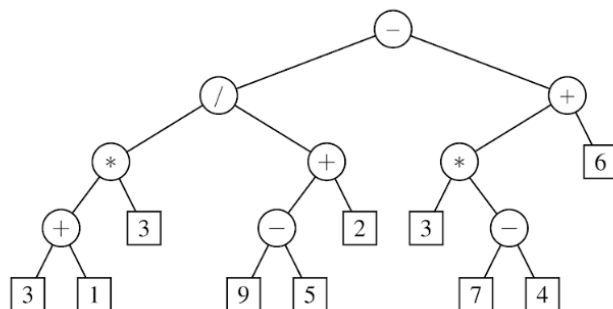
Queenslandi Ülikoolis (edetabelikoht 69) oli algoritmide ja andmestruktuuride kursus jagatud kaheks osaks, millest esimene oli „*Algorithms & Data Structures*“ ja teine „*Advanced Algorithms & Data Structures*“. Esimese osa ainukeseks soovitatavaks eeloscuseks oli nimetatud objektorienteeritud programmeerimise hea oskus ja teise osa alguses eeldati, et õpilane on omandanud kõik esimese osa teadmised ja on kursis diskreetse matemaatika baasteadmistega. Antud ülikool on liigitatud 3. liigi alla, kuna siin on eraldi aine tehtud lihtsalt kordamiseks ning seda ei tehta aine teises osas (sellist lähenemisviisi muudes kursustes polnud, näiteks MIT-i järgmine kursus keskendus edasijõudnud algoritmidele, kuid siin teine osa vaatas neid samu, mida vaadati Tartu Ülikoolis ja Edinburghi Ülikoolis ehk kahe ainega kokku läbiti peaaegu sama palju kui muidu ühes). [14-16]

Kursuse esimene aine õpetas Michael T. Goodrichi, Roberto Tamassia ja Michael H. Goldwasseri poolt kirjutatud „*Data structures and algorithms in Java*“ põhjal ja teemad läbiti sellega samas järjekorras ning samad alateemad. Üldiselt esimene pool ainek sarnanes Tartu Ülikooli ainele algoritmid ja andmestruktuurid. Alguses selgitati kuidas andmestruktuure üldiselt, räägiti nende omadusest ja kuidas neid läbida (kas ees-, lõpp- või keskjärjestuses). [14-15]



Joonis 12. Queenslandi Ülikooli näide puust.

Joonisel 12 on näidatud, kuidas on seletatud selle kursuse sees puu olemust raamatu näitel. Raamat on jagatud osadeks nagu sissejuhatus ja erinevad alamosad, eelnimetatud on omakorda jagatud peatükkideks ja need eraldi veel väiksemateks osadeks. Samuti on joonisel 13 näidatud, et ka selle kursuse käigus on näitena kasutatud Eestis levinud aritmeetilise avaldise puud. [14-15]



Joonis 13. Queenslandi Ülikooli näide aritmeetilise avaldise puust.

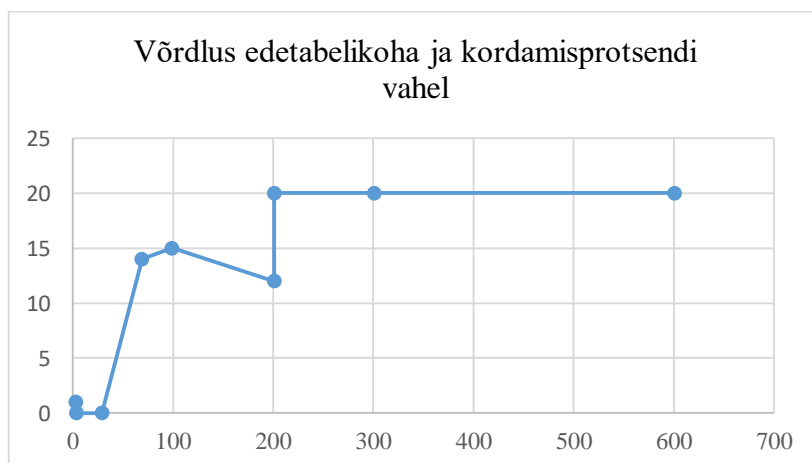
Pärast puude üldist käsitlemist oli juttu kahendpuudest, otsimispuudest ja tasakaalustatud puudest. Selgitati nende olemust ja nende peal rakendatavaid tegevusi (lisamine, eemaldamine ja tasakaalustamine) ning sellega lõppes 1. osas puude teema. [14-15]

Aine teine osa nimega „*Advanced Algorithms & Data Structures*“ sarnaneb väga Edinburghi Ülikooli uuritud kursusele, sest selle aine jooksul käsitletakse puid samuti vaid sorteerimis-algoritmide juures seletamaks jaga ja valitse algoritme nagu kiirsorteerimine ja mestimis-sorteerimine. Üldiselt oli selles aines eeldatud väga suuresti eelteadmisi. [15-16]

3 Analüüs

Analüüsi osas võetakse kokku uurimuslikust osast saadud info ja selle põhjal tehakse järeldusi, kuidas õpetatakse maailmas erinevatel algoritmide ja andmestruktuuride kursustel puude teemat ning üritatakse erinevate meetodite abil välja selgitada, mis võiks olla kõige optimaalsem viis seda teha.

Uurimuse üheks hüpoteesiks oli see, et mida kõrgemal edetabelikohal on ülikool maailma koolidega võrreldes, siis seda vähem keskendutakse seal kordavale materjalile. Selle uurimiseks kasutati edetabelina *The Times Higher Education World University Rankings 2019* edetabelit, kus on reastatud 1258 ülikooli paremuse alusel. Uuritud ülikoolidest ei leidunud edetabelis vaid Tallinna Ülikooli. Taltech on seal nimekirjas *Tallinn University of Technology* (ehk TTÜ) nime all, mida kasutas ülikool kuni 2018. aasta 17. septembrini. Kordavate teemade mahtu võrreldakse puude põhiteemade (sissejuhatus, omadused ja kahendpuud) materjali mahuga võrreldes kogu puude teemale pühendatud mahuga. Joonisel 14 võib näha antud võrdluse tulemusi.



Joonis 14. Võrdlus edetabelikoha ja kordamisprotsendi vahel.

Joonise horisontaalteljel on edetabelikohad ja vertikaalteljel on kordamisele kulutatav maht protsentuaalselt, mis on ümardatud. Kui Tallinna Ülikool oleks ka mahtunud tabelisse, siis oleks see graafik jätkanud sama sirget mööda kuni edetabelikohani, sest Eesti ülikoolid õpetasid kõik umbes sama mahuga uuritud materjale. Joonisel võib näha, et kaks ülikooli on saanud sama edetabelikoha, seda saab selgitada asjaoluga, et edetabel hakkas alates 201. kohast andma kohti vahemikuna ja mõlemad ülikoolid (Aucklandi Ülikool ja Waterloo Ülikool) said 201.-250. koha.

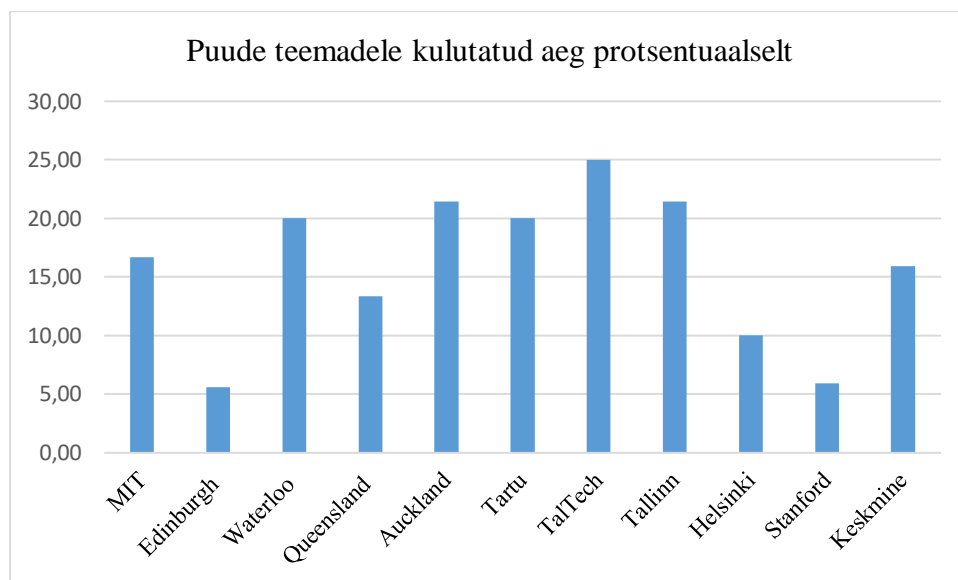
Kui vaadata saadud tulemusi, siis ei saa väita, et oleme töö alguses võetud hüpoteesi ümber lükanud, kuna jooniselt on näha, et edetabeli koha tõustes tõuseb ka kordamise maht protsentuaalselt, kuid esineb väikseid kõrvalekaldeid (Stanfordi ja Waterloo ülikoolid) ning uuritavate kursuste maht ei ole ka piisavalt suur, et väita midagi kindlat. Siiski võib graafikult näha, et on olemas mingisugune seos nende kahe parameetri vahel. Suhet iseloomustavaks valemiks valiti $y = 4.2041 \ln x - 6.1342$, kus y tähistab kordamise protsenti ja x tähistab ülikooli edetabelikohta. Valitud valem on logaritmiline, kuna see iseloomustab kõige paremini saadud tulemusi. Parima korrelatsioonikordaja saab, kui valida polünomiaalne valem, kuid sel juhul tekib graafikus tõuse ja languseid, mida selle uurimuse raames ei suudeta põhjendada ning sõltuvalt ka polünoomi astmest võib kordamise protsent minna negatiivseks, mis aga ei ole võimalik. Saadud andmete ja selle suhte (eelnevalt mainitud logaritmilise valemi) korrelatsioonikordaja on 0.8086. Korrelatsioonikordaja tugevust iseloomustatakse järgmiste vahemikega – 0-0.3 on olematu seos andmete vahel, 0.31 – 0.5 on nõrk seos, 0.51 – 0.7 on keskmise tugevusega seos ja kordaja suurem kui 0.71 näitab tugevat seost. Saadud kordaja (0.8086) näitab nende andmete põhjal, et leidub tugev seos uuritud ülikoolide edetabelikoha ja kordava materjali hulga vahel.

Uurimuse eesmärgiks oli ka saadud andmete analüüs ja optimaalse õpetamisviisi väljaselgitamine. Antud andmete põhjal võib eeldada, et algoritmide ja andmestruktuuride aines saavutatakse parim tulemus, kui ainele on eelnenud mingi sissejuhatav aine, kus on õpilasele selgitanud andmestruktuuride baasomadusi ning seeläbi peab AA aines vähem aega kulutama sellele ja saab keskenduda siis kas põhjalikumalt algoritmide selgitamisele või saab juurde võtta keerulisemaid teemasid.

Erinevatest kasutatud meetoditest üks ainulaadsem oli koodinäidete kasutamine loengumaterjalis õppevahendina. Seda kasutas Waterloo Ülikool oma materjalides korduvalt ning vähesel määral Helsingi Ülikool ja Stanfordi Ülikool kasutas oma materjalis pseudokoodi. Koodi kasutamine andmestruktuuride õpetamisel annab õpilasele parema arusaama sellest, mida ta päriselt kunagi rakendama hakkab. Tänu sellele on suurem tõenäosus, et kui peab lahendama ülesannet antud andmestruktuuri põhjal, siis suudab ta seda juba vähemalt baasteadmiste raames teha. Selle suurimaks veaks on aga suur ajaline maht. Selleks et selgitada õpilasele täpselt ära, mida kood teeb, võib minna rohkem aega kui muidu mõne muu visuaalse näitega (näiteks kui panna slaidile pilt kahendpuust ja kõrvale kood realiseeritud kahendpuust, siis õpilane mõistab pildi sisu tõenäoliselt varem) ja see võib takistada piisavalt kiiret edasiminekut aines. Kui aine on vaid teoreetiline, siis on koodinäited pigem kasulikud, kuid kui

aines on ka praktikumid (nagu olid kõigis uuritud ainetes), siis oleks parem hoida koodinäited selgitamise eesmärgil pigem sinna.

Üldiselt oli oodatav, et erinevates koolides õpetatakse ainet erinevat viisi ja igal moodusel on omad halvad ja head küljed ning sellepärast tulekski optimaalseima õpetusviisi leidmiseks kasutada erinevaid meetodeid erinevatelt ülikoolidelt. Näiteks selgus uurimuse käigus, et kõrgema edetabelikohaga koolid jätavad kordava osa minimaalseks, seega parima tulemuse saamiseks peaks juba eeldama, et õpilased on juba eelnevalt omandanud mingid teoreetilised baasteadmised teemadest ja kordamise jätma minimaalseks (ideaalis lihtsalt korra üle käia ja mitte neile keskenduda pikemalt). Seda saab saavutada näiteks täiendava aine lisamisega või eelnevas aines sissejuhatavate teemade läbimisega. Puude teemadele üldiselt kulutatav aeg kogu aine mahuga võrreldes erines suuresti mõnes ülikoolis, kuid see pigem jäi üle 10%.

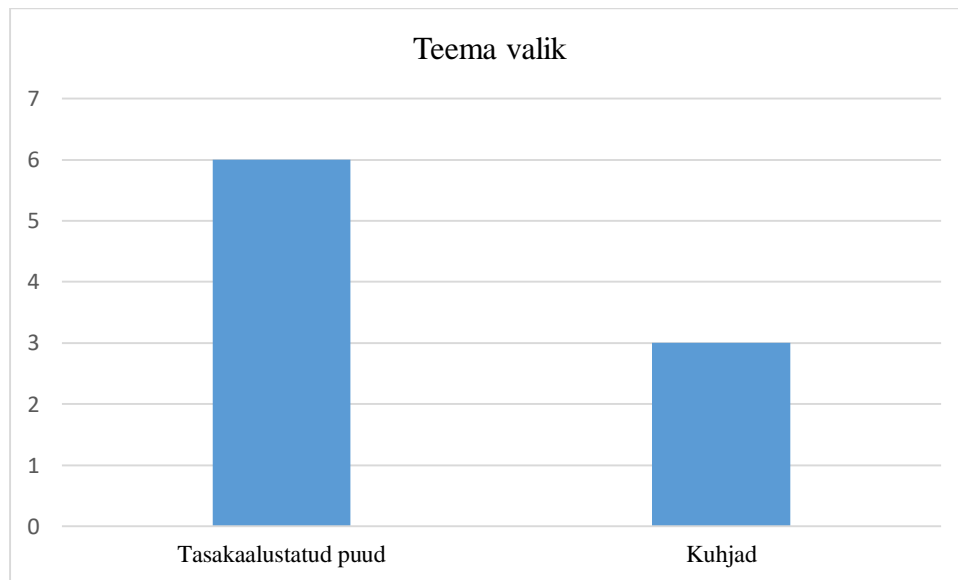


Joonis 15. Puude teemadele kulutatud aeg.

Jooniselt 15 on näha, et keskmiselt kulutati puude teemadele umbes 16% kogu ajast (arvutatud puid käsitletud loengute arv jagatud kogu loengute arvuga), selle viib alla Edinburghi Ülikooli ja Stanfordi Ülikooli väga madal protsent, ilma selleta läheneks keskmise protsent 18% peale. Järelikult võib andmetest järeldada, et kui aine kokku on 15 loengut, siis sellest umbes 3 peaksid käsitlema puid, mida tegi näiteks Tartu Ülikool.

Veel üheks erinevuseks ülikoolide ainete ülesehitusel oli teemade järjekord. Kui aines oli ruumi kordamisele, siis see oli alati esimese teemana võetud, kuid järgmised teemad varieerusid. Pärast kordamist tuli järgmise teemana kas tasakaalustatud puud või kuhjad. Siin ei ole ühte ja õiget lähenemist, kuna mõlemat viisi harrastasid nii edetabelis eespool olevad ülikoolid kui ka

tagapool olevad ülikoolid, seega siit ei saa teha järeldust, kumb variant oleks parem. Näiteks MIT (edetabelikoht 4) ja Tallinna Ülikool (ei mahtunud esimese 1258 ülikooli hulka) läbisid enne kuhjade teema, kuid Tartu Ülikool (edetabelikoht 301-350) ja Stanfordi Ülikool (edetabelikoht 3) läbisid tasakaalustatud puude teema varem. Joonisel 16 on kujutatud kuidas jaotusid teemade valikud ülikoolide hulga suhtes.



Joonis 16. Teema valiku võrdlus.

Kuna tasakaalustatud puude on valinud esimeseks teemaks kaks korda rohkem ülikoole, siis pigem tundub see olevat variant, mis on laialdasemalt levinud ja ehk ka sellevõrra võib tunduda inimestele loogilisema järjestusena.

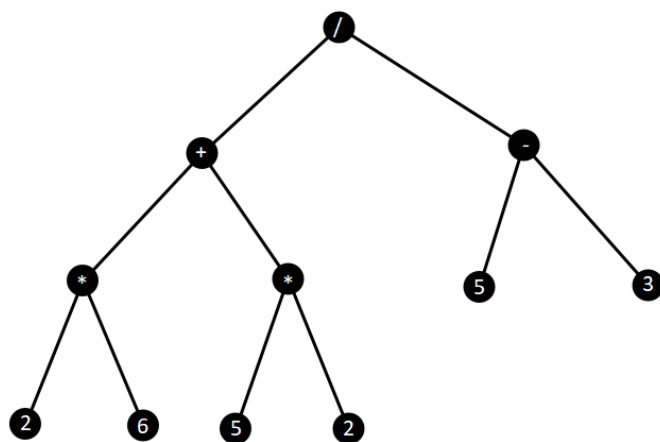
4 Näidisülesanded

Bakalaureusetöö üheks algselt püstitatud eesmärgiks oli välja selgitada, mis ülesandeid kasutavad maailma ülikoolid teatud teemade õpetamisel ja neist valida enim kasutust leidnud või mingil muul põhusel parimaks valitud ülesanded ning moodustada neist kogum, mis peaks olema abiks puude teema õpetamisel. Valitud ülesanded on jaotatud teemade kaupa gruppidesse (kordavad teemad, kahendotsimise puud, tasakaalustatud puud ja kuhjad).

4.1 Kordavad teemad

Väga mitmes kursuses kordavaid teemasid kas üldse ei kajastatud või tehti seda minimaalses koguses, kuid kuna Tartu Ülikoolis olid need teemad kajastatud, siis on ka valitud ülesannete hulgas mõni nendest teemadest. Kordavate teemade alla loetakse selle töö käigus puu (eelkõige kahendpuu) koostamine ja selle läbimine.

Kahendpuu koostamise õpetamiseks oli mitmes ülikoolis kasutatud näitena aritmeetilise avaldise puu ülesannet, kus anti ette mingisugune avaldis ja ülesandeks oli see õige kujuga puuks teha. Esimeseks näidisülesandeks on valitud avaldise $\frac{(2*6)+(5*2)}{5-3}$ puukujule viimise.

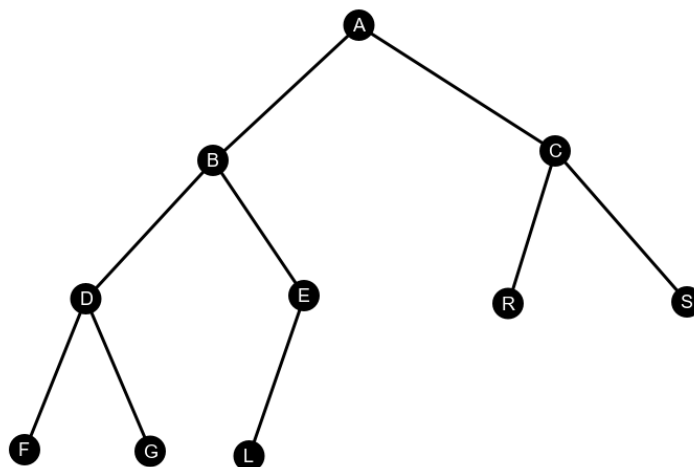


Joonis 17. Aritmeetilise avaldise puu ülesande näidislahendus.

Joonisel 17 on kujutatud ülesande vastus. Kui õpilane suudab seda ülesannet korrektselt lahendada, siis võib eeldada, et tal on olemas piisav arusaam puude omadustest ja kuidas neid rakendada (saab aru, milline tipp on juur, millised alamad teatud tipu juurde käivad jms).

Puu läbimise õpetamiseks on valitud ülesanne, kus joonisel 18 olev puu tuleb läbida igas järjestuses eraldi (vastus keskjärjestuses on FDGBLEARCS, eesjärjestuses on ABDFGELCRS).

ja lõppjärjestuses on FGDLEBRSCA). Kui õpilane suudab selle ülesande korrektselt lahendada, siis võib eeldada, et ta oskab läbida puid vajatavas järjestuses.



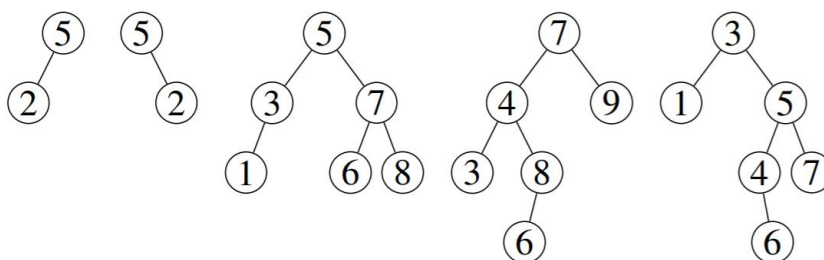
Joonis 18. Järjestuse ülesandes kasutatav puu.

Nende kahe ülesandega on terve kordav osa kaetud. Kui osata lahendada mõlemat ülesannet, siis on omandatud põhilised teadmised puudest ja selle omadustest ning osatakse vabalt puudel liigelda. Neid oskusi omades saab minna järgmise teema juurde, milleks on kahendotsimise puud.

4.2 Kahendotsimise puud

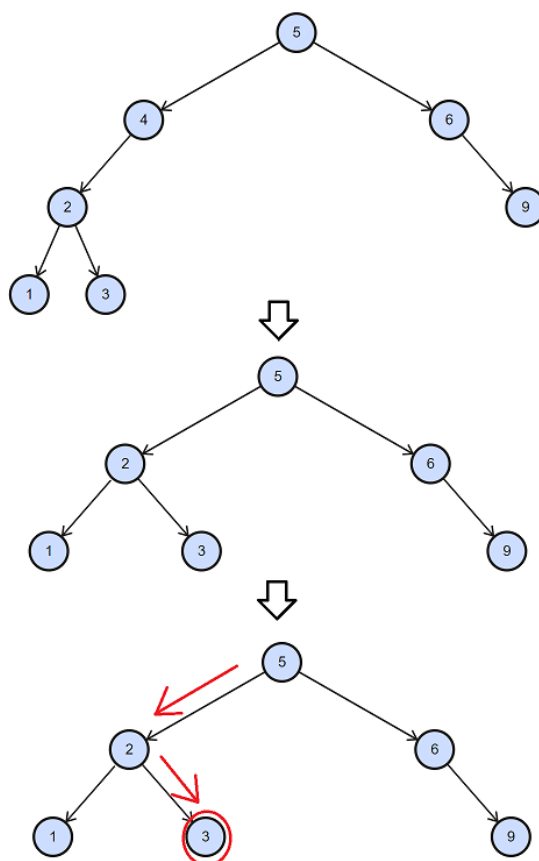
Kahendotsimise puud on teema, mis oli kajastatud peaaegu kõigis algoritmide ja andmestruktuuride kursustes. Kui kordamist eraldi ei tehtud, siis see teema võeti üldjuhul esimese teemana (erandina MIT), kuid kui oli ees eraldi kordamise osa, siis see teema tuli alati järgmisena. Kahendotsimise puude teemas peaks lugeja ära õppima, mis põhimõttel sellised puud töötavad ja kuidas neid kasutada. Selleks peab oskama kahendotsimise puid koostada ja kirjeid otsida, lisada ning eemaldada.

Üheks sissejuhatavaks ülesandeks sobiks ülesanne, kus on antud mitu erinevat kahendpuud ning nende hulgast peab leidma kahendotsimise puud ja selgitama, miks just need on otsimispuud. Selle ülesandega näidatakse, et saadakse aru kahendotsimise puude põhiomadustest. Näitena sellisest ülesandest on kasutatud Tartu Ülikooli ülesannete kogust võetud ülesannet (joonis 19), kus 5 erineva puu hulgast tuleb leida kahendotsimise puud. Vastus sellele ülesandele – otsimispuud on (vasakult paremale järjekorras) 1., 3. ja 5., sest 2. puul on juurtipu parem alam temast väiksem ja 4. juurtipu vasaku alama parem alam juurtipust suurema väärtusega ehk see tipp peaks olema juurtipust paremal pool.



Joonis 19. Kahendpuid määravate ülesannete kasutatavad puud. [3]

Teise ülesandena sobiks kompleksülesanne, mis koosneb kõigist kahendpuid peal tehtavatest operatsioonidest. Pakutud ülesanne on järgmine – kõigepealt tuleb koostada puu, mis koosneb kirjetest 5, 4, 6, 9, 2, 3 ning 1 ja sisestada need antud järjekorras. Pärast seda tuleb eemaldada kirje 4 ning alles jäänud puust tuleb otsida kirje 3. Ülesande visuaalne lahendus on joonisel 20.



Joonis 20. Kahendpuid operatsioonide ülesande näidislahendus.

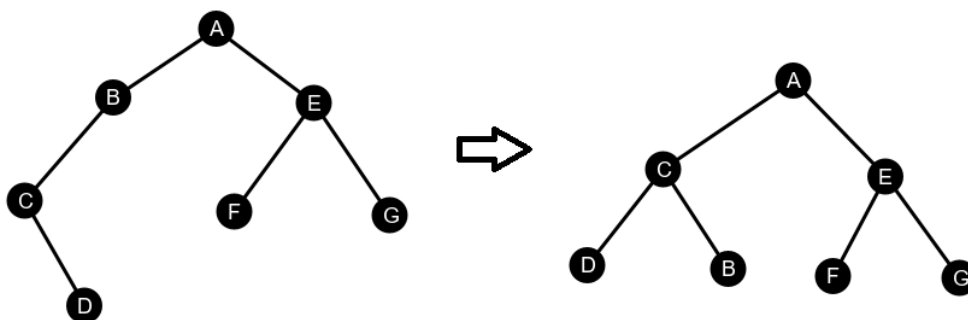
Kui osata eelmist kahte ülesannet, siis võib eeldada, et ollakse aru saanud kahendotsimise puu tööpõhimõttest ja osatakse nende peal sooritada erinevaid operatsioone. Kui esimene ülesanne esines uuritud kursustes pigem seal, kus esines rohkem kordamist, siis kahendpuid erinevate

operatsioonide ülesandeid leidis igas kursuses, kus seda teemat käsitleti ja sealt võib eeldada, et antud ülesanne on oluline teema üldiseks mõistmiseks.

4.3 Tasakaalustatud puud

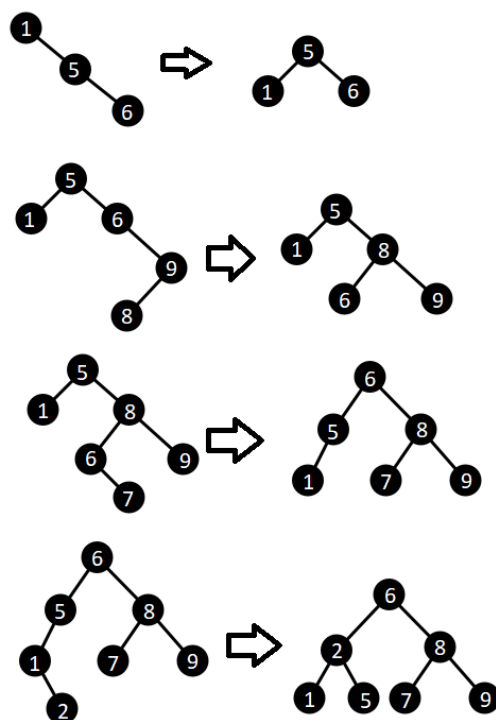
Pärast kahendotsimise puude selgestegemist on võimalik edasi liikuda nende tasakaalustamise juurde. Kui eelmises teemas selgitati juba lisamist ja otsimist, siis selles teemas tuleb selgeks teha, kuidas teha nii, et nende operatsioonide rakendamisel säiliks puu tasakaalustatud kuju (tasakaalu all on antud teemas mõeldud, et alamate maksimaalne kõrguste vahe võib olla 1) ehk tuleb selgeks teha puu peal tehtavad pöörded (parem, vasak, paremvasak ja vasakparem).

Esimese ülesandena on mõistlik alustada millegi kergemaga, näiteks kuidas leida puus tasakaalustamata koht ja see tasakaalustada. Näiteülesandena on kasutatud ülesannet, kus etteantud puus on juure vasak alampuu tasakaalust väljas ning selle korrastamiseks on vaja sooritada parempööre (joonis 21).



Joonis 21. Tasakaalustamise ülesande näidislahendus.

Teiseks ülesandeks on pakutud samuti kompleksülesanne, mis koosneb puu moodustamisest, kuid puu peab igal sammul olema tasakaalustatud ning ülesande käigus on vaja sooritada mitu pööret. Näidisülesandes tuleb puusse lisada 1, 5, 6, 9, 8, 7 ja 2. Kuna eemaldamisel ja lisamisel kehtib sama tasakaalustamise põhimõte, siis ei ole vaja eemaldamist eraldi teha. Ülesande lahendus on kujutatud joonisel 22, kus igat lisamist pole näidatud, eraldi on väljatoodud kohad, kus tuleb tasakaalustada.



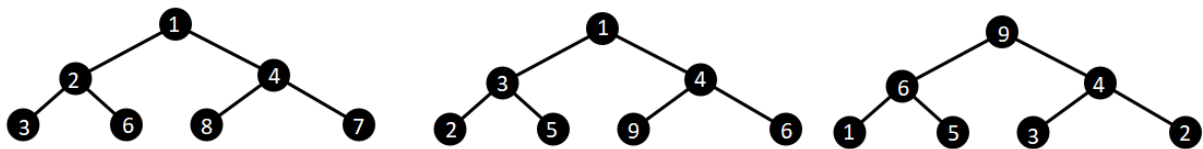
Joonis 22. Tasakaalustatud puu konstrueerimine.

Nende kahe ülesande lahendamise on õpilane tõestanud, et omab baasteadmisi tasakaalustatud puudest ja nende operatsioonidest ning on suuteline neid (vähemalt paberil) rakendada. Esimene ülesanne oli väga levinud uuritud kursuste seas, kasutati näiteks ka meetodit, kus pidi erinevate ülesannetena tegema kõik 4 pööret, kuid selline ülesanne, kus õpilane ei saa keskenduda vaid oma ühele tegevusele (nagu teine näidisülesanne), annab parema ülevaate sellest, kas on õigesti teemast aru saadud.

4.4. Kuhjad

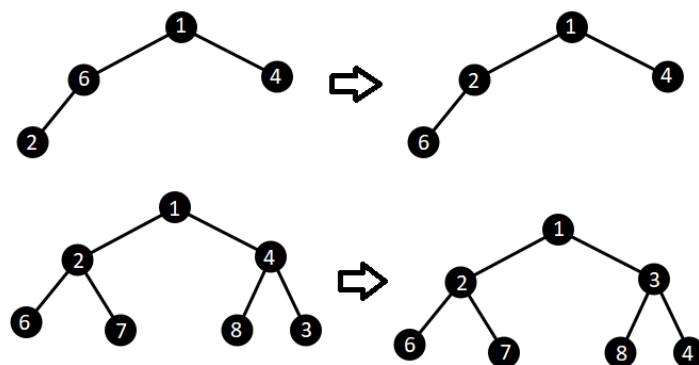
Kuhjade teemat kajastati erinevates kursustes erinevat viisi, mõnes ei olnud seda üldse teemaks, mõnes käidi ruttu läbi ja mõni peatus sellel pikemalt. Sellepärast on näidisülesanded valitud ka sellised, mis annaksid hea ülevaate teemast, kuid ei nõua väga palju aega ning kuna kursustes oli pigem keskendutud kahendkuhjadele, siis on ka näidisülesanded sellest teemast. Sellest teemast peaks õpilane oskama kuhjade omadusi ning kuidas kuhjastada.

Esimese ülesandena on väljapakutud ülesanne, kus mitmest kahendpuust on vaja valida, milline on kahendkuhi ja milline kahendpöördkuhi. Joonisel 23 on kujutatud 3 kahendpuud, millest esimene on kahendkuhi ja kolmas on kahendpöördkuhi. Teine kahendpuu ei ole kuhi, kuna tipu, mille kirje on 3, alluvaks on tipp, mille kirjeks on 2 ja see läheb kuhja omadusega vastuollu, mis lubab alluvateks vaid sama suure või suurema kirjega tippe.



Joonis 23. Kuhjade määramise ülesanne.

Teiseks ülesandeks on elementide kuhjastamine. Selline ülesanne eeldab teadmisi kuhjade omadustest ning oskust elemente kuhjas kas üles või alla liigutada. Kuhjastatavateks elementideks on 1, 6, 4, 2, 7, 8 ja 3. Joonisel 24 on kujutatud antud elementide kuhjastamist, iga elemendi eraldi lisamist pole kajastatud, on vaid need kohad, kus tuleb elementide asukohta kuhjas muuta.



Joonis 24. Kuhjastamise ülesanne.

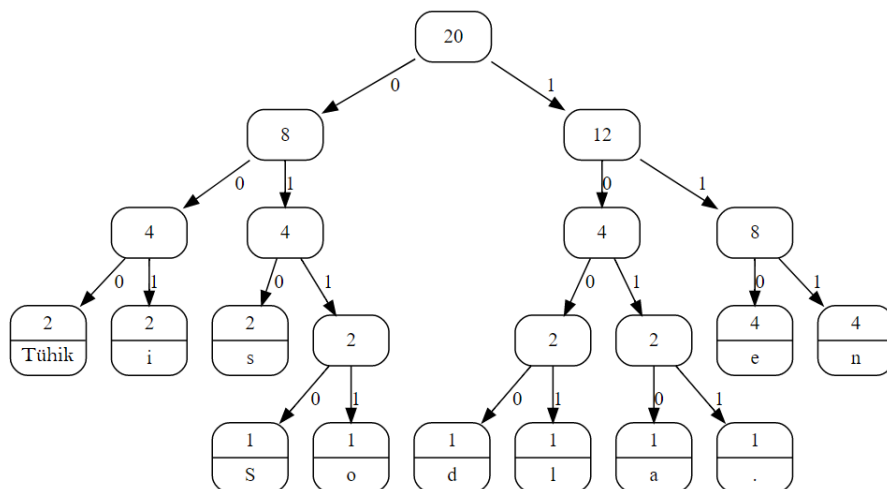
Nende kahe ülesandega tõestab õpilane, et omab baasteadmisi kahendkuhjadest ning oskab neid kasutada ülesannete lahendamiseks. Esimesele ülesandele sarnased ülesanded leidsid kasutust pigem kursustes, kus oli kordamist rohkem, kuid teise ülesande sarnaseid kasutati peaaegu igas kursuses, kus kuhjade teema oli sees.

4.5 Keerulisemad ülesanded

Eelmised ülesanded on oma raskusastmelt pigem kergemad ja neid saab kasutada antud teemade õpetamisel, kuid selles alateemas pakutud ülesanded on oma raskusastmel veidi keerulisemad ning neid võib kasutada näiteks kontrollimaks, kas teemast on piisavalt hästi aru saadud. Need ülesanded eeldavad õpilaselt rohkem kui baasteadmisi antud teemadest.

Väga mitmel kursusel oli juttu Huffmani puu koostamisest. Järgmises ülesandes tuleb koostada lause „See on prooviülesanne.“ põhjal Huffmani puu ning avaldada sealt bitijadana täht o. Huffmani kodeerimise põhimõtteks on tähed paigutada kahendpuus, mille alamate tähisteks on 1 ja 0 nii, et enam kasutust leidvad tähed saaksid kirjeldatud lühema bitijadaga, seega tuleb õpilasel kõigepealt leida tähtede esinemissagedus lauses ning seejärel hakata vähem esinenud

tähti kokku panema, kuni kõik on ühes puus. Üks võimalikke saadud puid on joonisel 25. Tähe o bitijada on antud puu põhjal 01111.



Joonis 25. Huffmani puu ülesanne.

Kuna siiani on nädisülesannete lahendamiseks piisanud vaid teoreetilisest oskusest puid kasutada, siis järgmine ülesanne on pühendatud oskusele oma teadmised puudest koodina kirja panna. Sarnast ülesannet on kasutanud ka näiteks Tartu Ülikool oma aine raames. Ülesandeks on kirjutada pseudokood, mis kontrollib, kas täielikus kahendotsimise puus on lehttippude puhul täidetud kahendotsimise puu tingimused ning tagastab vastava tõeväärtuse, puu tipu nimi on tipp, parema alluva nimi on parem, vasaku alluva nimi on vasak ja nende kirje on info. Lahendusena pakutud koodis (joonis 26) on pseudokoodi asemel kasutatud Pythoni keelt ning ülesande lahendamiseks on ka mitmeid muid võimalusi.

```
def kontroll(tipp):
    list = []
    if tipp.vasak.vasak == None and tipp.vasak.parem == None:
        if tipp.vasak.info < tipp.info: #vasak tipp peab olema väiksem
            list += [True]
        else:
            list += [False]
    if tipp.parem.vasak == None and tipp.parem.parem == None:
        if tipp.parem.info > tipp.info: #parem tipp peab olema suurem
            list += [True]
        else:
            list += [False]
    try:
        list.append(kontroll(tipp.vasak))
        list.append(kontroll(tipp.parem))
    except:
        pass
    if False in list:
        return False
    else:
        return True
```

Joonis 26. Täieliku kahendpuu lehttippude ülesande kood.

Sellised ülesanded õpetavad teoreetilisi teadmisi ka päriselt rakendama. Mõnes kursuses oli proovitud neid teadmisi juba õpetamise käigus anda (näiteks kasutas Waterloo Ülikool koodinäiteid teemade seletamisel), kuid enamus seda ei teinud ning sellised ülesanded on hädavajalikud, et õpilane suudaks reaalselt rakendada kursusest saadud teadmisi.

Raskemad ülesanded kontrollivad õpilase sügavamaid teadmisi teemast. Isegi kui õpilane pole kursis Huffmani kodeerimisega, siis kui ülesandes kirjeldada selle põhimõtteid, siis peaks ta oskama lahendada esimest ülesannet, kuna eelnevate teemade läbimisel on piisavalt selgitatud puude omadusi ja kuidas neid rakendada. Selles ülesandes on lihtsalt selleks antud kaasa mingi täiendav tekst. Teise ülesande puhul on tegu lihtsalt puu läbimisega ning mõistete tundmisega (mis on lehttipu ja täieliku puu), mis ei tohiks raskust valmistada, kui ollakse eelnevalt kursis sellega. Tingimus, et uurida tuleb vaid lehttippe, muudab ülesande keerulisemaks, kuna niiviisi peab õpilane näitama, et teeb vahet lehttipul ja tavalisel tipul ning nende omadustel.

5 Kokkuvõte

Käesoleva bakalaureuse töö üheks eesmärgiks oli uurida, kuidas õpetatakse puude teemat Tartu Ülikoolis, teistes Eesti ülikoolides ja kursustel üle maailma ning võrrelda saadud tulemusi ning tulemuste analüüsi põhjal välja selgitada kõige optimaalsem viis kuidas antud teemat õpetada aine Algoritmid ja andmestruktuurid raames. Teise eesmärgina pidi valmima kogu ülesannetest, mis peaks aitama kõige paremini õpilasele selgitada antud teemasid ning kontrollima, kas õpilane soovitud teadmised ka omandas.

Tulemuste analüüsi käigus leiti, et üheks oluliseks tingimuseks teema paremaks õpetamiseks oli vähene kordamine ja eeldamine, et õpilased on eelnevalt juba teemaga kursis. See võimaldaks minna teemadega piisavalt kiiresti edasi, et keskenduda rohkem kas olemasolevatele teemadele või võtta mõni teema lisaks juurde. Kui jätta kordavad teemad kas eelmisesse ainesse või lisada eraldi kordav aine, siis algoritmide ja andmestruktuuride aine saaks keskenduda keerulisematele teemadele või selgitada olemasolevaid teemasid põhjalikumalt, mis tagaks arvatavasti parema arusaama teemast.

Samuti leiti, et puude teema õpetamisele võiks kogu aine raamest kuluda ligi 20% kogu ajast, mida Tartu Ülikool hetkel juba oma õpetamises kasutab. Lisaks tuli erinevate kursuste põhjal välja ka, et teemade järjestus on kohati erinev, kuid see ei omanud erilist vahet, kuna polnud vahet, kumb viis valiti (kas tasakaalustatud puud enne kujade teemat või vastupidi), mõlema valiku oli teinud nii edetabeli eesotsas olnud ülikoole kui ka pigem tagapool olevaid koole ning kokkuvõttes oli ainukene vahe selles, kui paljud ülikoolid olid ühe või teise tee valinud. Populaarsem variant oli tasakaalustatud puud enne kuhjade teemat.

Ülikoolide kursuste uurimise ja analüüsi põhjal valmis ka komplekt ülesannetest, mis peaksid olema abiks puude teema õpetamisel ning kontrollimisel, kas õpilane on omandanud vajalikud teadmised.

Antud bakalaureusetöö edasiarendusena oleks võimalik uurida seost ülikoolide koha paremus-edetabelis ja kordamisele kulutatava aja vahel. Antud töö leidis uuritud ülikoolide edetabeli-koha ja kordava materjali vahel tugeva seose, kuid uuritud ülikoolide maht võis olla liiga väike, et see seos tõestada.

Viidatud kirjandus

1. Kiho, Jüri. Algoritmid ja andmestruktuurid. Tartu: 2003
2. Petuhhov, Inga. Puud üldiselt. Põhilised algoritmid puudel. (loengumaterjal) – vaadatud 03.01.2019, http://www.cs.tlu.ee/~inga/alg_andm/tree_gen_2018.pdf
3. Peder, Ahti, Kiho, Jüri, Nestra, Härmel. Algoritmid ja andmestruktuurid ülesannete kogu. Tartu: 2017
4. Pöial, Jaanus. ICD0001 - Algoritmid ja andmestruktuurid – vaadatud 23.04.2019, <http://enos.itcollege.ee/~jpoial/algoritmid/>
5. The Times Higher Education World University Rankings 2019 – vaadatud 20.04.2019, https://www.timeshighereducation.com/world-university-rankings/2019/world-ranking#!/page/0/length/-1/sort_by/rank/sort_order/asc/cols/stats
6. Morris, John. Data Structures and Algorithms – vaadatud 23.04.2019, https://www.cs.auckland.ac.nz/software/AlgAnim/ds_ToC.html
7. Harder, Douglas Wilhelm. Algorithms and Data Structures, Lecture Materials – vaadatud 20.04.2019, https://ece.uwaterloo.ca/~dwharder/aads/Lecture_materials/
8. Demaine, Erik, Devadas, Srin. Introduction to Algorithms, Lecture notes – vaadatud 20.04.2019, <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/lecture-notes/>
9. Laaksonen, Antti. Tietorakenteet ja algoritmit – vaadatud 03.05.2019, <https://courses.helsinki.fi/en/tkt20001/124962188>
10. Laaksonen, Antti. Tietorakenteet ja algoritmit –kirja – vaadatud 03.05.2019, <https://www.cs.helsinki.fi/u/ahslaaks/tirakirja/>

11. Stanford University. Design and Analysis of Algorithms – vaadatud 03.05.2019,
<http://web.stanford.edu/class/cs161/>
12. Mayr, Richard. Algorithms and Data Structures – vaadatud 20.04.2019,
<http://www.inf.ed.ac.uk/teaching/courses/ads/>
13. Shimodaira, Hiroshi. Informatics 2B – vaadatud 20.04.2019,
<http://www.inf.ed.ac.uk/teaching/courses/inf2b/>
14. Thomas, Richard. Algorithms & Data Structures – vaadatud 20.04.2019,
https://www.courses.uq.edu.au/student_section_loader.php?section=1&profileId=87127
15. Goodrich, Michael T., Tamassia Roberto, Goldwasser Michael H. Data structures and algorithms in Java – vaadatud 20.04.2019, <https://uq.rl.talis.com/items/C825E880-1311-C4B6-2B58-6AE444C19CB4.html?referrer=%2Flists%2FEE90887E-9F97-1C13-EFDA-DAB4398CAB70.html%23item-C825E880-1311-C4B6-2B58-6AE444C19CB4>
16. University of Queensland. Advanced Algorithms & Data Structures – vaadatud 20.04.2019, https://www.courses.uq.edu.au/student_section_loader.php?section=1&profileId=35952

Lisad

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Karl-Johannes Kalma,

(autori nimi)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

Puude käsitusviisid ja õpetamine,

(lõputöö pealkiri)

mille juhendaja on Ahti Peder,

(juhendaja nimi)

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Karl-Johannes Kalma

10.05.2019